多策略增强花授粉算法及其应用 *

李大海，伍兆前，王振东

(江西理工大学 信息工程学院，江西 赣州 341099)

摘　要：针对经典花授粉算法容易陷入局部最优解和收敛速度慢的缺点，提出一种增强型透镜成像策略和随机邻域变异策略的花授粉算法。通过增强型透镜成像策略扩展花授粉算法的搜索空间，增加解的多样性有助于算法跳出局部最优解。引入随机邻域变异策略，借助邻域内的信息指导算法搜索，增强算法的收敛精度和搜索速度。并且对改进后的花授粉算法和 4 种其他改进算法在 CEC2013 测试函数上比较，实验证明改进后的多策略花授粉算法不管是在收敛精度还是搜索速度都比对比算法优秀。最后把多策略花授粉算法应用在汽车传动参数模型上研究该算法的实际效用，结果表明多策略花授粉算法在汽车传动参数优化问题上都优于对比算法。

关键词：随机邻域变异；透镜成像；花授粉算法；参数优化；收敛精度

中图分类号：TP301.6　　doi: 10.19734/j.issn.1001-3695.2022.02.0036

Multi-strategy flower pollination optimization algorithm for vehicle power transmission parameters

Li Dahai, Wu Zhaoqian, Wang Zhendong

(*School of Information Engineering*, *Jiangxi University of Science & Technology*, *Ganzhou Jiangxi* 341099, *China*)

Abstract: Classic flower pollination algorithm (FPA) can be easily exposed to the shortcomings of local optimal solution and slow convergence velocity. In view of these shortcomings, this paper proposed an FPA with an enhanced lens imaging strategy and random neighborhood-based mutation strategy. The lens imaging strategy can help the algorithm avoid the shortcoming of local optimal solution by expanding the search space of FPA to increase the diversity of the solution. The introduction of random neighborhood-based mutation strategy can enhance the convergence accuracy and search speed of the algorithm by guiding algorithm search with information in the neighborhood. A comparison of the improved FPA with four other improved algorithms on CEC2013 test function found that the improved multi-strategy FPA performs better than the comparison algorithms in both convergence accuracy and search speed. To study its practical utility, this paper applies the multi-strategy FPA into the automobile transmission parameter model and the results indicate that multi-strategy FPA is better than the comparison algorithm in optimization of automobile transmission parameters.

Key words: random neighborhood variation; lens imaging; flower pollination algorithm; parameter optimization; convergence accuracy

0　引言

花授粉算法(Flower Pollination Algorithm, FPA) [1]是由 Yang 于 2012 年提出一种简单高效的元启发式群智能优化算法。该算法将植物的异花授粉和自花授粉的过程类比到算法中的全局搜索和局部搜索，平衡全局搜索和局部搜索之间的切换。由于 FPA 算法具有结构简单、参数少、鲁棒性和适应性强等特点，其已经应用于求解诸多领域的复杂优化问题求解，如无线传感

器网络布置优化[2]，混凝土 3D 框架优化[3]，医学图像分割[4]，切比雪夫多项式神经网络优化[5]。但 FPA 算法也有着和其他群智能优化算法相同的问题，容易陷入局部最优解并且收敛速度慢[6]，国内外学者针对 FPA 算法已经提出诸多的改进措施。文献[7]提出一种结合克隆选择策略的改进 FPA 算法其使用随机游走替代 FPA 算法中使用的 Levy 飞行并且在局部授粉之前克隆最优解，以提高最优解的接受概率。文献[8] 提出了一种结合模拟退火策略的 SFPA 算法，该增强 FPA 算法利用模拟退火策略使算法避免陷入局部最优解，从而提高了算法的全局搜索能力。文献[9]中提出了一种基于精英对立策略的 EOFPA 算法该算法采用了全局性精英对立和局部自适应的贪婪策略，其中全局性精英对立的学习增强了群体的多样性，局部自适应的贪婪策略增强了其探索能力。文献[10]中提出了一种结合差分进化变异策略的 ssFPA/DE 算法，该新算法将 FPA 算法的搜索策略特征和差分进化的变异策略相结合从而增强算法的搜索效率。文献[11]中针对 FPA 算法易陷入局部最优，后期收敛速度慢的缺点，提出一种基于引力搜索机制的 GSFPA 算法，通过花朵个体间的引力和本身算法的 Levy 飞行共同实现个体位置的更新，使花朵受 Levy 飞行和个体间引力的双重影响，个体之间通过共享优化信息提高算法的寻优能力。文献[12]中提出 FA-FPA 算法，在 FPA 算法的局部授粉过程中引入自适应的变异因子，并对 FPA 算法的概率转换进行自适应调整。最后将其与萤火虫算法相结合，实验表明 FA-FPA 算法具有更高的收敛性和稳定性。

上述对于 FPA 改进的算法主要集中在三点：一是借用策略某方面的优越性弥补 FPA 算法的自身缺陷。例如模拟退火的突跳机制使其避免陷入局部最优解；引力搜索机制调整个体位置更新。二是增加种群多样性。例如全局性精英对立学习增强种群多样性。三是融合其他算法，借用其他算法的优秀机制增强 FPA 算法。

尽管目前已经提出了诸多的改进 FPA 算法，但 FPA 算

收稿日期：2022-02-09；**修回日期：**2022-03-25　**基金项目：**国家自然科学基金资助项目(61563019)；国家自然科学基金资助项目(61562037)

作者简介：李大海(1975-)，男，山东乳山人，副教授，硕导，博士，主要研究方向为分布式系统服务质量、强化学习算法、粒子群算法；伍兆前(1993-)，男，江西南昌人，硕士研究生，主要研究方向为博弈论、粒子群算法(894085330@qq.com)；王振东(1982-)，男，副教授，博士，主要研究方向为无线传感器网络节点覆盖、定位与路由、人工智能及网络安全.

法仍存在诸多的改进空间。本文提出一种基于多策略的 MSFPA 算法，其主要从以下两方面改进 FPA 算法：

a) 通过增强型透镜成像策略去增强算法的种群多样性。透镜成像策略是一种基于光学原理的反向学习策略，该策略能够增强种群的多样性改善算法的搜索范围，帮助算法跳出局部最优解。本文提出一种增强型透镜成像策略，以 10 次迭代为一回合。在标准透镜成像生成反向解的基础上，再利用算法生成的最优解和透镜成像的反向解两者之间的笛卡尔积生成精英集合，对精英集合内部的有序对分别求适应值，精英集合内部始终保持 10 个个体，新的优质解会替代劣质解。

b) 引入差分进化算法的邻域变异策略。由于邻域变异策能够平衡全局和局部搜索，所以本文提出将随机邻域变异策略和 FPA 粉算法的搜索策略相结合，并且为了适应 FPA 算法的全局搜索和局部搜索的随机切换，将变异因子 F 改为自适应，减小不合适的变异因子对算法的影响程度，最后借用邻域集合内部的信息指导搜索，加快收敛速度和收敛精度。

通过对比 FPA，TMFPA，HLFPA，ISSA 四种算法在 CEC2013 测试函数上的性能，验证本文提出算法的有效性，并把算法应用到汽车传动系统参数优化的工程问题上，进一步验证算法优化复杂问题的能力。

1　花授粉算法概述(FPA)

FPA 算法借鉴自然界花授粉的过程，基本模拟了自然界花的自花授粉和异花授粉的行为。授粉的媒介分为两种：生物和非生物。生物授粉的过程视为异花授粉，一般由动物或者昆虫充当载体，可在大范围内传播；非生物的自花授粉视为自花授粉，通常是借助风实现传播。为了更好利用 FPA 算法解决优化问题，Yang 在算法中假定每一株植物只开一朵花，每朵花只有一个花粉胚子，一个胚子对应优化问题中的一个解。此外还需假定以下 4 条规定：

1) 植物的异花授粉对应算法的探索阶段，全局搜索通过生物充当载体，并遵循 Levy 飞行的规律来实现；

2) 非生物的自花授粉对应算法的开发阶段，局部搜索是同种类的植物不同花朵之间通过风实现传粉；

3) 繁衍概率对应花的特性，两朵花(个体)的相似和关联性成比例；

4) 概率转换参数 $p\in[0,1]$ 对 FPA 算法的探索(全局授粉)和开发(局部授粉)之间的相互转换进行调节。由于受到位置、风等因素影响，整个授粉过程中，局部授粉的概率更大。

由上述的描述可知，异花授粉(全局授粉)和自花授粉(局部授粉)为 FPA 算法的核心。FPA 算法的全局授粉按式(1)进行：

$$x_i^{t+1}=x_i^t+\gamma L\times(x_i^t-g),rand>p \tag{1}$$

其中，x_i^{t+1}，x_i^t 为第 $t+1$ 和 t 代的解，g 为全局最优解，γ 是步长控制因子，L 是 Levy 飞行的主步长[13]，步长计算公式如下：

$$L\approx\frac{\lambda\Gamma(\lambda)\sin(\frac{\pi\lambda}{2})}{\pi}\times\frac{1}{s^{1+\lambda}},s>s_0>0 \tag{2}$$

其中，λ =3/2，$\Gamma(\lambda)$ 是标准的伽马函数。由于按式(1)更新 x_i^t，所以整个全局搜索阶段都是朝着当前的全局最优解 g 的方向探索，并使用 Levy 飞行每一次都随机调整步长，以确保整个种群的多样性。通过此种方式获取的每一代的最优解 g 充当下一次迭代的基向量，保证最优解基因的传承。

FPA 的局部授粉按如下式(3)进行：

$$x_i^{t+1}=x_i^t+\varepsilon\times(x_j^t-x_k^t),rand\le p \tag{3}$$

其中，ε 是在[0,1]区间上服从均匀分布的随机数，x_j^t,x_k^t 代表与 x_i^t 不同的随机选取的花粉。其目的是确保算法在局部搜索阶段具有较好的开发性，且用 ε 扰动确保每次获取的解都具有随机性。

2　多策略花授粉算法(MSFPA)

FPA 算法作为一种群智能优化算法，同样临如何平衡算法种群多样性、收敛速度和收敛精度的问题。本文提出利用增强型透镜成像反向学习策略和基于随机邻域变异策略两项措施改进 FPA 算法，以进一步加强算法的收敛速度和精度。

2.1　增强型透镜成像反向学习策略

凸透镜成像规律是一种光学定律。如图 1 所示，假设物体 P，在 x 轴上的投影为 x，高度为 h，通过凸透镜可以获得一个实像 P^*，设实像 P^* 在 x 轴上的投影为 x^*，设投影 x 和 x^* 在 x 轴上的投影落于区间 $[a,b]$ 之内，则 x 和 x^* 满足如式(4)所示的关系：

$$\frac{(a+b)/2-x}{x^*-(a+b)/2}=\frac{h}{h^*} \tag{4}$$

令 $n=\frac{h}{h^*}$，并称 n 为缩放因子。对式(4)进行变换即可得到 x^* 的计算式(5)：

$$x^*=\frac{a+b}{2}+\frac{a+b}{2n}-\frac{x}{n} \tag{5}$$

若将 x 作为一维优化问题的某个解(或者解的坐标)，则 x^* 可以视为解 x 的反向解。特别的，当 n =1 时将，式(5)可简化成式(6)的形式，其就是一般的反向学习策略。

$$x^*=a+b-x \tag{6}$$

在透镜成像中可以通过调整缩放因子 n 的大小来动态获取某个解的反向解，对一般的 D 维优化问题，设 D 维向量 x 为该优化问题的某个解(或者解的坐标)。设 x_j 为当向量 x 的第 j 维分量，按式(7)得 x_j 的反向分量 x_j^*，则可得以 x_j^* 作为的第 j 维分量 D 维向量 x^* 作为解 x 的透镜反向解。

$$x_j^*=\frac{a_j+b_j}{2}+\frac{a_j+b_j}{2n}-\frac{x_j}{n} \tag{7}$$

其中，a_j,b_j 为搜索空间在第 j 维的上下界。

综合以上的透镜成像原理，本文提出一种增强型的透镜成像策略。首先将开始 10 次迭代生成的当前最优解保存在 S_1 集合中，将其由透镜成像生成的反向解保存在集合 S_2 中。并生成合 S_1 和 S_2 的的笛卡尔乘积集合 Q，如式(8)所示。

$$Q=S_1\times S_2=\{(x_1,x_2)\mid x_1\in S_1\ and\ x_2\in S_2\} \tag{8}$$

在随后的算法迭代中，如果当前迭代次数不是 10 的倍数，则将 S_1，S_2 记录的个体的适应值分别和当前最优解及其反向解的适应值进行对比并更新集合 S_1 和 S_2。每隔 10 次迭代，再按式(8)重新生成集合 Q。

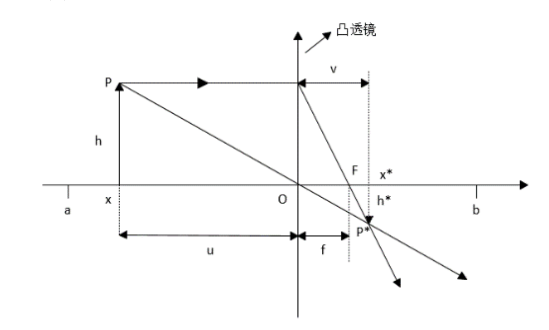


图 1　透镜成像

Fig. 1　Lens imaging

集合 S_1 和 S_2 分别记录了当前已知的适应值最好的 10 个解和其反向解，而集合 Q 作为 S_1 和 S_2 记录了最多 100 个由 S_1 和 S_2 的解构成的有序对。

由于透镜成像中的 n 是控制者原生解和镜像之间的距离，较小的 n 生成的镜像范围更大，而较大的 n 生成的镜像范围更小。通过上述的改进之后固定的 n 值已经很难适应 FPA 算

法全局和局部搜索随机转换的机制，因此需要改变 n 取值方式，具体公式如下：

$$n=(1+(t/T)^{1/2})^{k} \tag{9}$$

其中，t 代表当前迭代次数；T 代表最大迭代次数；k 为控制常数，控制曲线变化趋势。因此 k 值能够控制缩放因子的大小，也就是控制原生解和镜像解之间的距离，所以为了精确控制常数 k 的取值，选取 Rastrigin's 函数作为对比不同 k 值得测试函数，如图 2 所示。3 次对比，k 分别取 5、10、15。通过图 3 可以看出当 k 取 10 时效果最佳。

n 作为缩放因子迭代初期，n 值被放大镜像的取值范围就被放大，算法的探索区域也随之变大，有利于算法的全局搜索和探索速度。随着迭代次数的增加 n 也随之变小，相应的镜像取值范围也变小，加强算法的搜索精度，同时也有利于帮助算法跳出局部最优解。其次如果算法迭代到后期，当前轮和上一轮获取的解有可能会一样，那么存储在 S_1 中解会重复。但是通过自适应收缩因子 n 计算不同轮的反向解是不一样的，也就避免了重复存储的弊端。

相较于基本的透镜成像策略，增前型透镜成像策略改变了每一次迭代生成当前解的反向解模式。而是改为每十次迭代为一回合，通过把每次迭代生成的解和反向解存入 S_1 和 S_2 集合中，并且按式(8)生成集合 Q 。集合 Q 与 S_1 和 S_2 的关系就是一种映射，可以通过集合 Q 中的二元组映射到 S_1 和 S_2 内的个体。通过把集合 Q 内部二元组映射的解与反向解和当前解与反向解作对比，胜出者作为更新种群的向量。相较于基本的透镜成像策略的每一轮生成一个当前解的反向解，增强型透镜成像策略可以每一回合生成 100 个有序对，通过搭配不同轮的反向解找寻到最合适的反向解。因此增强型透镜成像策略在原有的基础上进一步扩大了搜索空间，有利于算法的多样性。并且对比透镜成像策略的固定缩放因子 n ，本文的增强型透镜成像策略采取自适应的缩放因子，使得策略可以更好的适应整个搜索过程。

为了对比增强型透镜成像和透镜成像策略的优劣性，以基本 FPA 算法为基础，选取经典测试函数 Sphere、Matyas、Rastrigin's 和 Ackely's 作为参照，详细内容如表 1 所示，适应值如图 3 所示。

表 1　测试函数

Tab. 1　Test function

测试函数	维度	范围	最优解	峰值
$f(x)=\sum_{i=1}^{n}x_i^2$	30	[-100,100]	0	单峰
$f(x,y)=0.26*(x^2+y^2)-0.48*x*y$	30	[-100,100]	0	单峰
$f(x,y)=-20+x^2+y^2-10(\cos 2\pi x+\cos 2\pi y)$	30	[-5.12,5.12]	0	多峰
$f(x,y)=-20\exp(-0.2\sqrt{0.5(x^2+y^2)})-\exp(0.5(\cos(2\pi x)+\cos(2\pi y)))+20+e$	30	[-30,30]	0	多峰

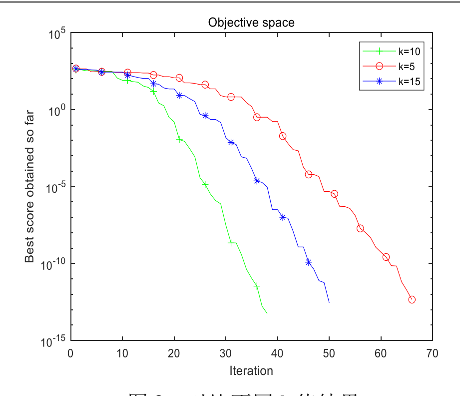


图 2　对比不同 k 值结果

Fig. 2　Compare the results of different k values

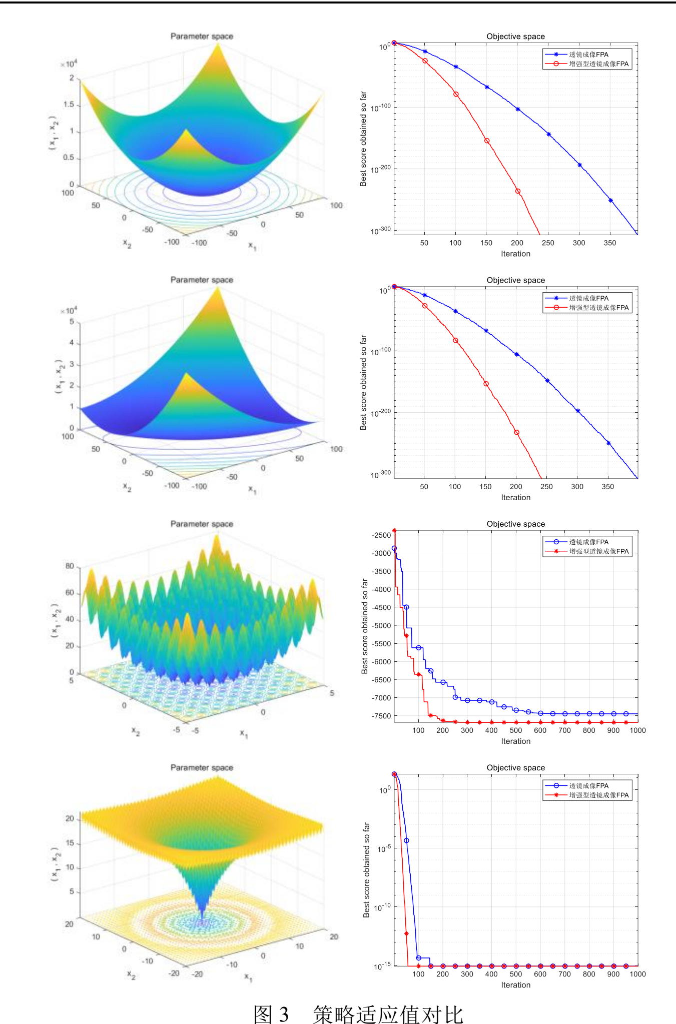


图 3　策略适应值对比

Fig. 3　Comparison of policy adaptation values

从图 3 中可以很明显的看出，在四个测试函数上，增强型透镜成像 FPA 算法比透镜成像 FPA 算法有更好的收敛性和收敛速度。

2.2　基于随机邻域变异策略的搜索方式

在解决复杂问题时，随着算法进入探索后期，一旦陷入局部最优解，FPA 算法往往很难跳出局部最优。一般的解决方案是在搜索过程中尽量增加候选解的多样性。如混沌映射[14]，模拟退火[15]等。近年，因差分进化算法(DE) [16]的变异策略的高效性和参数简单已经被许多学者用来和其他算法结合。DE 算法使用的基本的变异策略有两种：“DE/rand/1”和“DE/best/1” [17]。在前者策略中基向量从种群中随机选择，这意味具有更好的探索性，但开发性较差。后者则获取种群中最好的候选解作为基向量，从而具有较好的开发性，但探索性相对较差。Das[18]进一步提出了基于邻域的变异策略(DE/neighbor/1)，该策略引入邻域的信息指导搜索，以加快收敛的速度和收敛的精度。对于种群中的每个个体在每一代，DE/neighbor/1 随机的从群体中选择邻域集合，并以基础向量中当前个体邻域中的最好一个作为基向量。DE/neighbor/1 变异策略可以表述为如下：

$$V_i=x_{nbest}+F\times(x_{r1}-x_{r2}) \tag{10}$$

其中，x_{nbest} 为邻域内最优个体；x_{r1} 和 x_{r2} 为邻域内剩余个体中随机挑选的个体。

本文提出了一种基于 DE/neighbor/1 的改进变异策略，并将其应用于 FPA 算的局部搜索。改进变异策略表述如下：

$$rand<q: x_i^{t+1}=x^{*}+rand(x_j^t-x_i^t)+rand(x_i^t-x_k^t) \tag{11}$$

$$rand>q: x_i^{t+1}=x_{nbest}^t+F(x_j^t-x_i^t)+F(x_i^t-x_k^t) \tag{12}$$

其中，$q\in[0,1]$ 的常数，x_i^{t+1},x_i^t 代表第 $t+1,t$ 次的值，x^{*} 全局最优解，x_{nbest}^t 是邻域内最优解。对于式(11)中 x_j^t,x_i^t,x_k^t 都是从个体

中随机选取的，而式(12)中 x_j^t, x_i^t, x_k^t 都是从邻域内随机选取的个体。

对于邻域的划分以当前解为中心，按欧氏距离由小到大选取邻域集合，n 维的欧氏距离公式如下所示。

$$d=\sqrt{\sum_{j=1}^{n}(x_{m,j}-x_{n,j})^2} \tag{13}$$

需要注意每一代邻域内有数量限制，而算法迭代过程中的 x_i 选择合理的邻域个体数量 N_i，对平衡全局和局部搜索能力，提高算法寻优能力起着重要作用。受文献[19]启发，邻域数量按如下公式设置：

$$N_i=N_{lb}+(N_{ub}-N_{lb})\times\frac{f(x_i)-f_{\min}}{\sum_{j=1}^{NP}(f(x_j)-f_{\min})} \tag{14}$$

其中: N_{lb} 和 N_{ub} 分别代表 N 的下限和上限， $f_{\min}$ 代表当前种群适应值的最小值， $f(x_i)$ 为个体 i 当代的适应值。

算法初期，种群中优质个体数量较少，能够寻得全局最优的概率较小，所以前期需要加强全局搜索能力，以获得较多的优质个体，提高全局寻优的能力。根据式(14)可知，初期 $\sum_{j=1}^{NP}(f(x_j)-f_{\min})$ 远大于 $f(x_i)-f_{\min}$，N_i 趋向于 N_{lb}。此时算法具有较好的全局搜索能力，从而加速搜索能力。随着迭代次数的增加，种群中优质个体不断增加，$\sum_{j=1}^{NP}(f(x_j)-f_{\min})$ 和 $f(x_i)-f_{\min}$ 均趋向于 0，N_i 趋向于 N_{ub}，此时该算法具有较好的局部开发能力，从而根据邻域信息寻找最优解。

式(10)相较于差分进化算法中的单个向量的策略，使用两个差分向量的变异策略可以提高种群的多样性，并且能够帮助算法跳出局部最优解。而且使用全局最优解作为当代的基向量，有利于解的优良基因被继承。而式(12)中的 F 代表变异因子，是一个实常数因数，控制差分向量缩放的作用。对于算法而言 F 如果取固定值的话太小或者太大都会导致算法过早收敛。因此使用自适应变异因子[20]去避免不合理的参数设置带来的影响，具体公式如下：

$$F=\begin{cases}F_{\max}-(F_{\max}-F_{\min})\times\dfrac{f_i-f_{\min}}{f_{\max}+f_{\min}}, & f_i\ge\bar{f}\\ F_{\max}+\text{rand}\times F_{\min} & f_i<\bar{f}\end{cases} \tag{15}$$

其中：$F_{\max}$ 和 $F_{\min}$ 分别为 F 的上下限；$rand$ 是[0,1]之间的随机值；f_i 为个体 x_i 的适应度值；$f_{\max}$ $f_{\min}$ 为当前种群中最大最小的适应度值；$\bar{f}$ 为当前种群的平均适应值。

式(15)在文献[20]的自适应因子的基础上作出改动，当 $f_i<\bar{f}$ 时不在是固定的 $F_{\max}$，而是一个以 $F_{\max}$ 为下限的随机值，增加算法的随机性。自适应变异因子能够平衡算法寻优能力，一定程度上降低算法对参数的敏感程度，保证收敛速度和避免早熟。

为了进一步验证策略的有效性，把加入随机邻域变异策略的 FPA 和标准 FPA 算法进行比较，测试函数如表 1 所示，对比图如 4 所示。

图 4 中 Sphere 和 Matyas 函数分别选取了第 100 代和第 300 代的解作为对比，Rastrigin's 和 Ackely's 选取的是第 800 和第 1000 代的数据作为对比。选取的对比函数包含单峰和多峰，充分对比策略对于不同类型函数的适应性。需要注意的是随机邻域变异 FPA 在 Sphere 和 Matyas 函数上只有一个点，因为两次取值点重合了，而等高线上没有 FPA 的点是由于 FPA 算法获取的解超出了范围，也就是说随机邻域变异 FPA 在第 100 次迭代的时候已经收敛到 0 附近了，而同一时期的 FPA 算法离最优点很远。对于 Rastrigin's 和 Ackely's 函数而言分别取了第 800 代和第 1000 代作比较，可以看见两次取值都是不如随机邻域变异 FPA 算法。

通过自适应邻域数量策略，动态调整种群邻域的的数量，MSFPA 充分利用种群中"随机"信息和邻域内"最优"信息，合理的平衡了算法的全局和局部搜索能力，加速了信息的交换速度，增加种群的多样性，避免算法陷入局部最优解的同时，加速了算法的寻优速度，而且没有引入其他参数，算法的复杂度没有增加。

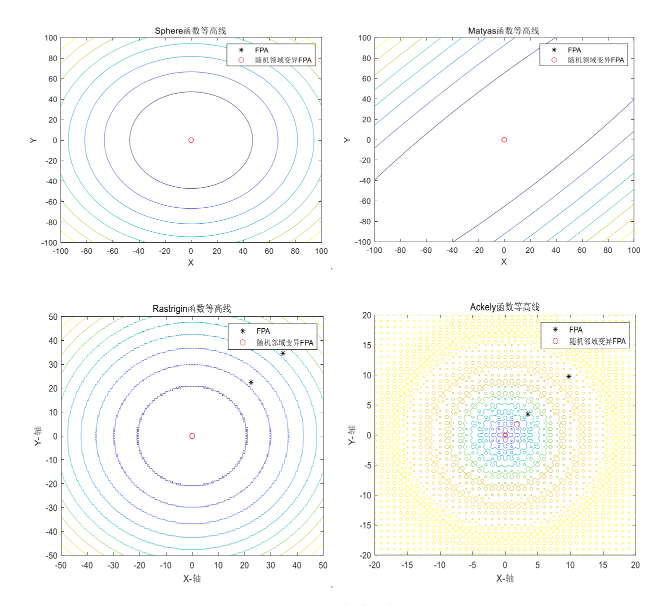


图 4　策略等高线对比

Fig. 4　Strategic contour comparison

2.3　MSFPA 的全局和局部搜索能力分析

算法的全局和局搜索能力通常作为衡量算法性能的两个指标。全局搜索代表算法的前中期的收敛能力，依靠全局搜索算法可以收敛到全局最优解大致的位置。而局部搜索是算法后期提升算法的搜索精度，能够让算法计算出的解无限接近全局最优解。FPA 算法依靠概率转换因子 p 去切换全局和局部搜索，全局搜索通过 Levy 飞行增加算法的随机性有利于算法的前期前中期探索能力，最后通过式(1)更新种群，有利于算法朝着全局最优解方向探索。局部搜索通过式(3)更新种群，从种群中选择最好的候选解作为基向量，保证算法的局部搜索阶段具有较好的开发性。而 MSFPA 算法引入增强型透镜成像，通过统计 10 轮的当前解以及对应反向解的排列组合后选择出全局最优解。相较于透镜成像只是对比当前解和反向解，增强型透镜成像策略拥有更多的候选解供其选择，增加算法的全局搜索能力以及种群的多样性，并且通过对缩放因子 n 采取自适应的方式控制当前解和反向解的距离，使其适应整个搜索阶段。另外增加了随机邻域变异策略，在花授粉的局部搜索阶段，对比常数因子 q 和 $rand$ 决定何种方式更新种群。式(12)因为使用全局最优解作为基向量，使其偏向于全局搜索的方式，保证了算法即使在局部搜索阶段也保持一定的收敛速度；式(13)使用是邻域集合作为基向量，并且集合内部的数量是会随着迭代变化更好的适应算法前期收敛速度快，后期收敛精度高的需求。并且配合自适应的变异因子控制算法的平衡性，使得 MSFPA 算法有着更好的综合寻优能力。

2.4　MSFPA 算法的时间复杂度分析

设需要优化的目标函数为 $f(x)$，解空间的维度为 d，根据 FPA 算法的步骤，FPA 算法的时间复杂度为 $O(d+f(d))$。根据 MSFPA 算法步骤，种群规模为 n，初始化个体位置的时间复杂度为 $O(nd)$，根据初始位置生成适应值的时间为 $f(d)$。全局搜索中基于 Levy 飞行生成步长的时间为 ξ_1，根据当前位置产生下一代时间为 ξ_2；局部搜索中的变异策略产生随机

数的时间为 ξ_3，根据当前位置产生下一代的时间为 ξ_4，设置变异因子 F 的时间为 ξ_5，设置邻域集合的时间为 ξ_6。

全局搜索的时间复杂度为

$$O(nd)+O(n(f(d)+\xi_1 d+\xi_2 d+f(d)))=O(d+f(d)) \tag{16}$$

局部搜索的时间复杂度为

$$O(nd)+O(n(f(d)+\xi_3+\xi_4+\xi_5+\xi_6+f(d)))=O(d+f(d)) \tag{17}$$

假设得到新的适应值与当前适应值比较的时间为 σ_1，经过比较后作为当前适应值的时间为 σ_2，后续生成精英集合以及对比过后生成新适应值作为全局适应值的时间为 σ_3。则算法总的时间复杂度为

$$2*O(d+f(d))+O(n(\sigma_1+\sigma_2+\sigma_3))=O(d+f(d)) \tag{18}$$

可以看到 MSFPA 与 FPA 算法的时间复杂度属于同一个数量级别，增强型透镜成像以及随机邻域的搜索策略并没有增加算法的时间复杂度。

改进的 MSFPA 算法伪代码如下所示。

算法 1 MSFPA 算法

```
输入：目标函数 f(x)；种群规模 NP；转换概率 p 和概率因子 q；变异因子 F 上下限( F_max, F_min )，最大迭代次数 MaxIter 。
输出：全局最优解。
初始化种群，在初始化种群中找到全局最优解 x* 。
for t=1: MaxIter
  for i=1: NP
    if  rand > p  进行全局授粉阶段
      绘制服从式(2)分布的随机步长向量 L
      按照式(1)对 x_i^{t+1} 位置进行更新
    else 进行局部授粉
      if  rand < q
        在当前种群中随机选取 3 个个体
        按照(12)对 x_i^{t+1} 位置进行更新
    else
      if 当前解的适应值 >=  mean(适应值)
        按式(16) f_i >= f̄ 部分设置变异因 F
      else
        按式(16) f_i < f̄ 部分设置变异因子 F
      end
      按式(15)设置邻域大小 N
      按式(14)计算距离，选取离当前最优解最近的 N 个解
      在当前种群中随机选取 3 个个体
      按式(13)对 x_i^{t+1} 位置进行更新
    end
  end
  按式(10)定义缩放因子
  for i=1: NP
    按式(8)生成当前解的反向解并进行越界处理
    对比当前解和反向解的适应值，最小的作为    当前解的适应值
  end
  if t<10
      把当前解存入 S_1 集合
      把当前解的反向解存入 S_2 集合
  else if mod(t,10) ~= 0
      把当前解和当前解的反向解与 S_1 和 S_2 中的解对比，淘汰劣质解
  else if mod(t,10) == 0
      得到 S_1 和 S_2 下标的有序对集合 Q
      for j = 1:size( Q ,1)
        求 Q 集合二元组对应 S_1 和 S_2 内部个体的适应值并记录两者适应值之和最小的一组有序对为最优组
        if 最优组第一元素的适应值 > 最优组第二元素的适应值
          第二元素映射的 S_2 中个体作为当前解
        else
          第一元素映射的 S_1 中个体作为当前解
        end
      end
  end
  对比种群个体， x* =最优个体，作为全局最优解 x*
```

3 实验结果和分析

为了研究多策略花授粉算法的有效性，选取 CEC2013 测试函数集进行测试，依次对 30/80 维度进行测试，各算法独立运行 10 次，记录最优值、平均值、标准差并与基本的花授粉算法和其他改进算法进行对比。参与对比的算法如表 2 所示，测试函数基本信息如表 3 所示，适应度曲线和最优解分布如图 5 和 6 所示。测试函数分为三类：F1-F5 单模态函数，F5-F12 为多模态函数,F13-F15 为复合函数。

表 2 对比算法

Tab. 2 Comparison algorithm

算法	算法简称
花授粉算法	FPA
三重动态调整的花授粉算法	HLFPA[21]
t-分布扰动策略和变异策略的花授粉算法	TMFPA[22]
多策略融合的麻雀算法	ISSA[23]

表 3 测试函数 CEC2013 说明

Tab. 3 Test function CEC2013 description

Type	Name	$f_i^*=f_i(x^*)$
Unimodal Functions	F1:Sphere Function	-1400
	F2:Rotated Discus Function	-1100
	F3:Different Powers Function	-1000
Basic Multimodal Functions	F4:Rotated Rosenbrock's Function	-900
	F5:Rotated Ackley's Function	-700
	F6:Rotated Weierstrass Function	-600
	F7:Rastrigin's Function	-400
	F8:Roatetd Rastrigin's Function	-300
	F9:Non-Continuous Rotated Rastrigin's Function	-200
	F10:Schwefel'sFunction	-100
	F11:Lunacek Bi_Rastrigin Function	300
	F12:Rotated Lunacek Bi_Rastrigin Function	400
Composition Functions	F13:Composition Function 2 (n=3,Unrotated)	800
	F14:Composition Function 4 (n=3,Unrotated)	1000
	F15:Composition Function 8 (n=3,Unrotated)	1400

注意：上述函数均来自 CEC2013 测试集

单模态函数 F1-F5：主要特征是在搜索区间内只有一个局部最小值。

多模态函数(F6-F12):这些函数具有多个局部最优值，但是与每个函数相对应的单个全局最优值。经典导数的优化算法在此类函数上不能取得满意效果，因为它们会陷入局部最优解。此类别的 7 个函数有以下特征：在任何地方都是不可分离的、不对称、连续的，但仅在一组点上是可区分的；在任何地方都是连续的，但在任何地方都是可微分的；具有从局部最优到全局最优的狭窄谷，具有大量的局部最优值，并且最佳和次佳最优解距离较远。

复合函数(F13-F15)：每个复合函数均由多个基本函数组成。3 个复合函数都是多模态，不可分离的和部队称的，并

且围绕不同的局部最优具有不同特征。

在实验中，每个函数都有一个相对应的全局最优值，该最优值不会随着问题的维度变化。实验所有函数的可变范围固定为$[-100,100]^p$，本次实验对 15 个函数测试每种算法，并且上述函数反复随机运行 10 次。

表 4、5 分别给出了 30 维、80 维的实验数据。从表 4 中数据可以看出 F1-F15 的函数 MSFPA 都是第一，这意味着该算法在整个搜索过程中寻优的精度和稳定性都要好于对比函数并且在其他对比算法搜索已经停滞的时候 MSFPA 算法依旧在向下探索说明基于笛卡尔积的透镜成像策略和随机邻域的变异策略可以帮助 MSFPA 算法跳出局部最优解。在 F13-F15 复合函数上均是排名第一，这说明 MSFPA 算法在处理复杂问题的寻优能力也强于对比函数并且对比其他算法 MSFPA 算法的收敛速度和收敛精度都要更高。在 30 维测试情况下，MSFPA 算法的 Ave Rank 为 1，排名第一。这表明融入多策略后该算法的低维度问题寻优能力得到了提升，并且搜索精度方面得到了加强。

表 5 列出了算法在 80 维测试函数上的测试数据。基本和 30 维数据相似，除了 F5 和 F10，其他的单模态，多模态和复合函数上都优于对比算法，最后的 Ave Rank 值为 1.33，排名第一。其 Ave Rank 值和 30 维接近，说明维度提升并没有降低该算法的性能。

图 5 是在 30 维的情况下，各类算法的收敛情况，可以明显看出，列出的图集中在单模态，多模态和复合函数上，其收敛精度都要优于大部分的改进算法。MSFPA 利用透镜成像和随机邻域的变异策略扩展搜索范围和改进搜索精度。例如 F5、F6、F7、F8、F9 和 F10，其他函数都基本停滞的时候，但是 MSFPA 算法还在向下搜索。这表明改进后确实对算法的寻优精度有显著提升，而且该算法的收敛速度对比其他的算法也相对较快。

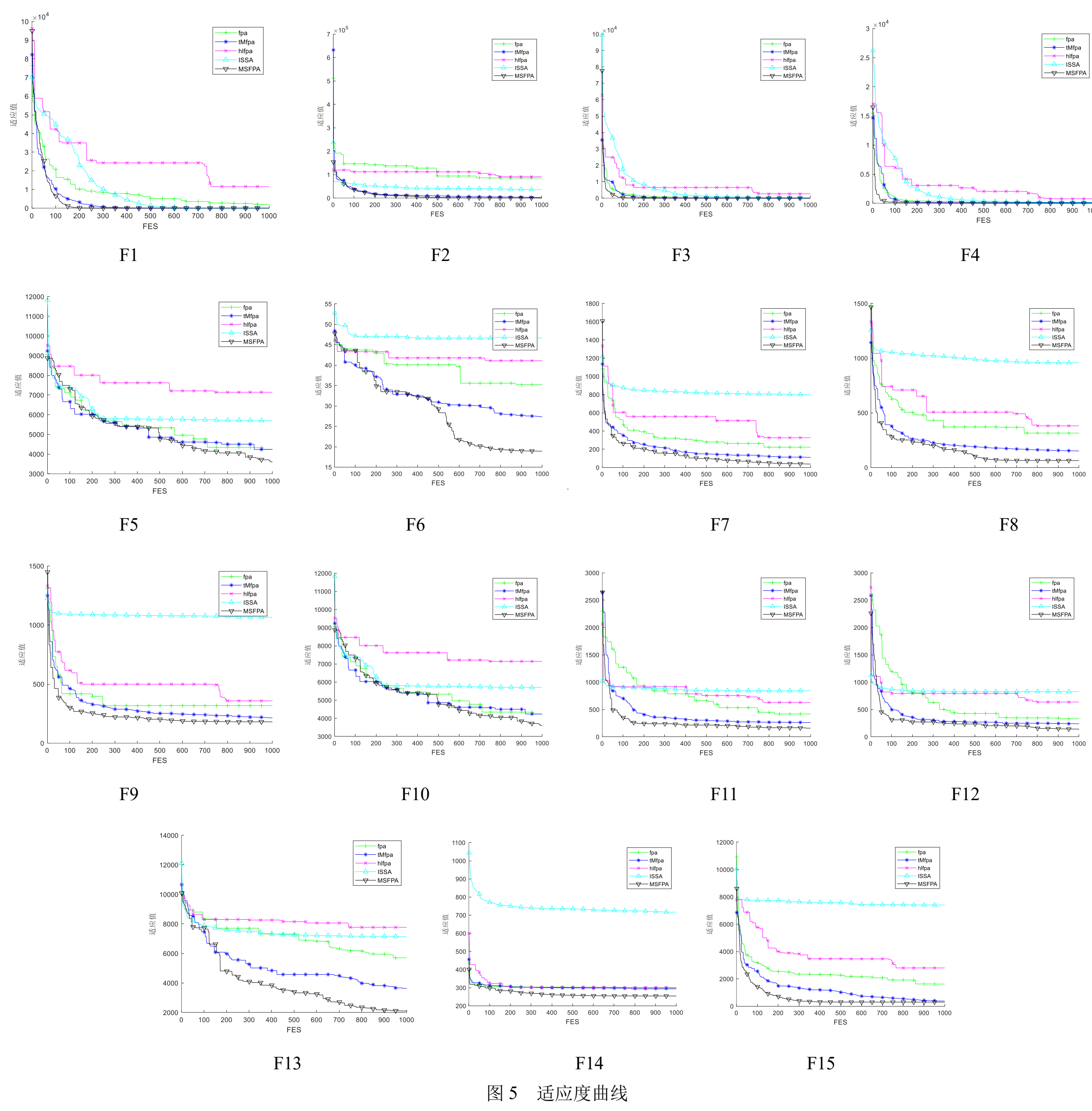


图 5　适应度曲线

Fig. 5　Adaptability curves

图 6 是所有算法 1000 次迭代获取最优解的箱线图，F1、F6、F8、F10 和 F13 中的 MSFPA 箱体较宽，代表该算法的所有最优值波动情况大，也就是算法收敛速度较快导致每一代的最优解之间跨度较大。其他的测试函数中箱体比较窄，代表算法从开始搜索到迭代结束获取的所有解变化不大。从图 7 可以看出 MSFPA 在 F1、F2、F6、F7、F8、F11、F13、F15 中箱体的下限比其他算法更低，也就代表它的搜索精度更高。

在 15 个测试函数中，30 维有 15 个，80 维有 13 个测试函数是优于 FPA、HLFPA、TMFPA、ISSA。这可以说明增强

型透镜成像策略和随机邻域的变异策略有助于跳出局部最优解，并能够指导算法后续的探索。

3.1 Friedman 检验

为了验证 MSFPA 算法和其他算法的显著性差异，采用 Friedman 检验[24]对算法进行参数检验。Friedman 检验是一种非参数双向方差检验方法，一般用来检测数据之间的显著性差异，多用来做单目标之间的优劣比较。一般实现步骤如下：

a) 收集实验数据。列出每个问题 i 的最好与最差结果排名，$r_i^j(1\le K\le j)$。

b) 求出所有算法的平均排名，并计算出最终排名。$R_j=(1-n)\sum_{i=1}^{n}r_i^j$。秩平均值越小则代表算法性能越好。

Friedman 统计值计算公式如(12)所示，F_f 的值越小，各算法之间的差异性就越大，当 n 和 k 足够大时(一般 n>10,k>5)，它是服从 k-1 自由度的 χ^2 分布。

$$F_f=\frac{12n}{k(k+1)}[\sum_j R_j^2-k(k-1)^2/4] \tag{19}$$

将检验 MSFPA 算法和对比算法在 30 维，80 维的情况，检验结果如表 6 所示。表中的 P-value 表示渐进显著性，用来判断是否存在显著性差异的重要指标，如果小于 0.01，则表示各项数据之间存在显著性差异。表中其他值表示各算法的秩平均值。表中的 30、80 维的 P-value 值明显小于 0.01。这表明 MSFPA 算法和其他对比算法存在显著性差异，这种差异有可能是多策略带来的解的多样性以及搜索能力的加强。各算法的秩平均值中，MSFPA 在每个维度都是最小的，因此 MSFPA 算法在各维度上的优化能力都得到了提升。

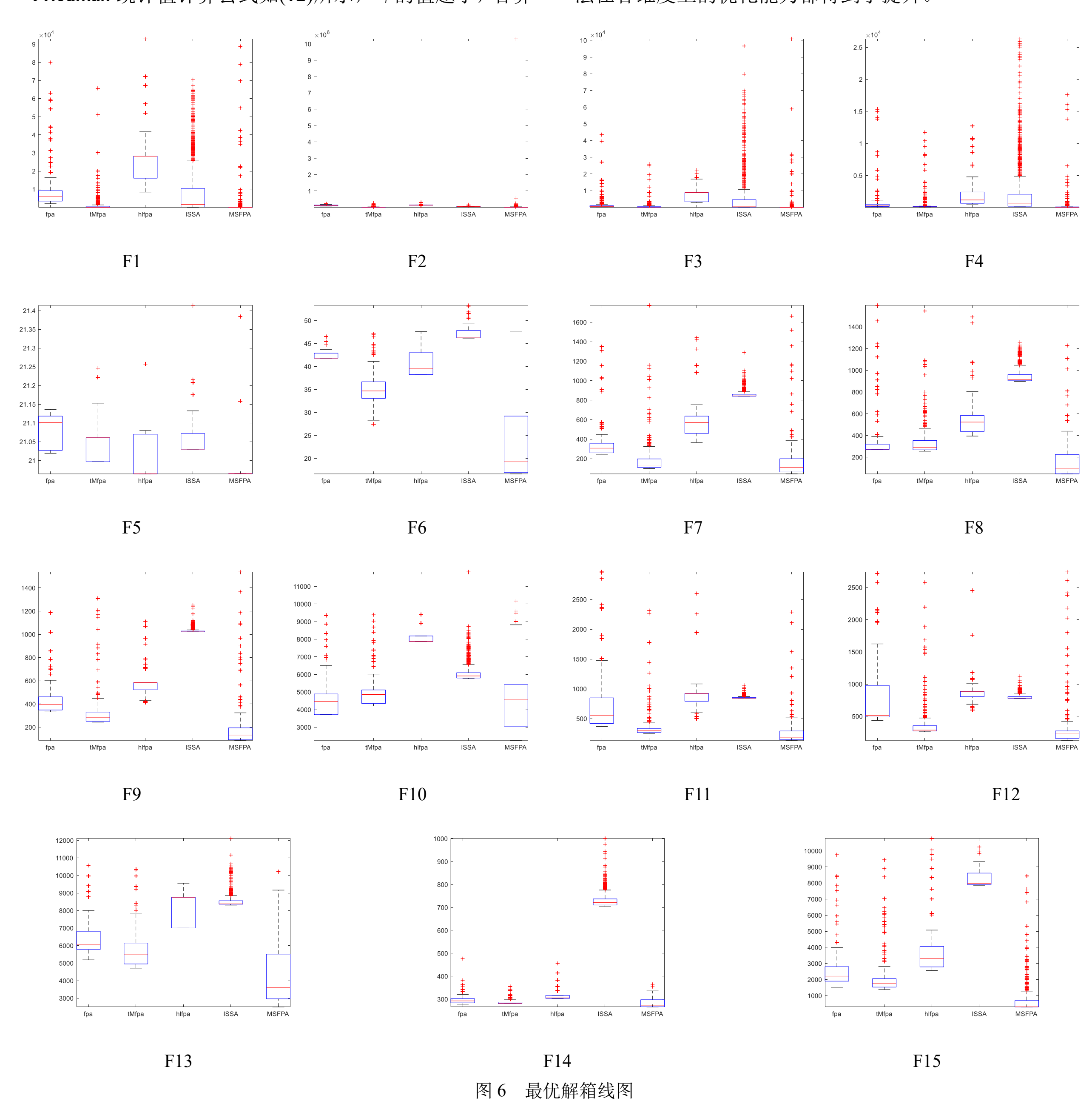


图 6　最优解箱线图

Fig. 6　Optimal solution boxplot

4　汽车动力传动参数优化

由内燃机理论可知，汽车的动力性和燃油经济性指标是相互矛盾的。即无法脱离汽车的动力性而单纯追求燃油经济性，也无法摆脱燃油经济性追求动力性，因此只能在汽车的动力性和燃油经济性选择一个平衡方案。而汽车动力性燃油经济性的综合评价体系和指标包括：

a) 动力性能发挥程度的评价指标为驱动功率损失率。

b) 经济性能发挥程度的评价指标为有效效率利用率。

c) 汽车动力传动系统匹配的综合指标为汽车能量利用率。

优化模型的设计变量选为

$$X=[X_1,X_2,X_3,X_4,X_5,X_6]^T=[i_{g1},i_{g2},i_{g3},i_{g4},i_{g5},i_0]^T \tag{20}$$

其中：i_{gj} 为变速器第 j 档的传动比($j=1,2\ldots5$)；$i_{g1}\in[1,5]$，$i_{g2}\in[1,4]$，$i_{g3}\in[0.5,3]$，$i_{g4}\in[0.5,2]$，$i_{g5}\in[0.3,1]$，$i_0\in[2,6]$ 为主减速器传动比。

表 4　实验结果

Tab. 4　Experimental results

测试函数	Mean/Std/Rank	FPA	TMFPA	HLFPA	ISSA	MSFPA
$f1$	Mean/Std	1.60E+03/2.68E+02	1.00E-01/9.50E-02	8.58E+03/1.78E+03	9.93E+00/1.88E+01	3.30E-12/1.91E-12
	Rank	4	2	5	3	1
$f2$	Mean/Std	9.24E+04/2.07E+04	3.13E+03/1.66E+03	7.78E+04/1.65E+04	3.76E+04/5.23E+03	1.32E+02/1.25E+02
	Rank	5	2	4	3	1
$f3$	Mean/Std	1.42E+02/4.18E+01	7.15E-01/3.46E-01	2.24E+03/3.25E+02	1.80E+02/5.37E+01	5.93E-08/4.15E-08
	Rank	3	2	5	4	1
$f4$	Mean/Std	1.36E+02/2.79E+01	4.34E+01/2.77E+01	5.71E+02/1.27E+02	1.48E+02/3.56E+01	2.01E+01/1.88E+01
	Rank	3	2	5	4	1
$f5$	Mean/Std	2.10E+01/6.17E-02	2.10E+01/6.63E-02	2.10E+01/4.96E-02	2.10E+01/4.45E-02	2.10E+01/7.18E-02
	Rank	3	2	4	5	1
$f6$	Mean/Std	3.90E+01/1.53E+00	2.58E+01/5.09E+00	3.95E+01/1.99E+00	4.40E+01/2.71E+00	2.21E+01/4.25E+00
	Rank	3	2	4	5	1
$f7$	Mean/Std	2.55E+02/2.84E+01	1.09E+02/2.70E+01	3.54E+02/2.95E+01	8.40E+02/7.62E+01	4.29E+01/1.39E+01
	Rank	3	2	4	5	1
$f8$	Mean/Std	2.96E+02/2.21E+01	1.97E+02/3.95E+01	3.87E+02/2.92E+01	9.38E+02/2.41E+01	6.94E+01/2.94E+01
	Rank	3	2	4	5	1
$f9$	Mean/Std	3.16E+02/2.54E+01	2.44E+02/3.47E+01	3.71E+02/2.02E+01	1.03E+03/8.36E+01	1.27E+02/2.75E+01
	Rank	3	2	4	5	1
$f10$	Mean/Std	4.38E+03/6.58E+02	4.14E+03/5.15E+02	7.23E+03/2.82E+02	5.86E+03/7.56E+02	2.89E+03/5.20E+02
	Rank	3	2	5	4	1
$f11$	Mean/Std	4.01E+02/5.63E+01	2.47E+02/5.85E+01	5.70E+02/3.49E+01	8.35E+02/1.89E+01	1.23E+02/1.52E+01
	Rank	3	2	4	5	1
$f12$	Mean/Std	3.94E+02/4.29E+01	3.03E+02/4.46E+01	5.77E+02/5.26E+01	8.16E+02/3.97E+01	1.47E+02/3.86E+01
	Rank	3	2	4	5	1
$f13$	Mean/Std	5.20E+03/9.38E+02	4.56E+03/3.92E+02	7.49E+03/4.03E+02	7.64E+03/7.81E+02	3.73E+03/7.74E+02
	Rank	3	2	4	5	1
$f14$	Mean/Std	2.81E+02/6.23E+00	2.72E+02/7.28E+00	2.95E+02/5.77E+00	7.53E+02/4.09E+01	2.64E+02/7.69E+00
	Rank	3	2	4	5	1
$f15$	Mean/Std	1.79E+03/1.42E+02	1.09E+03/1.02E+03	2.35E+03/1.86E+02	7.71E+03/1.08E+03	4.19E+02/3.75E+02
	Rank	3	2	4	5	1
Count		0	0	0	0	15
Ave Rank		3.2	2	4.2	4.53	1
Total Rank		3	2	4	5	1

维度:30，以上数据独立运行 10 次的均值

表 5　实验结果

Tab. 5　Experimental results

测试函数	Mean/Std/Rank	FPA	TMFPA	HLFPA	ISSA	MSFPA
$f1$	Mean/Std	5.70E+04/1.13E+04	1.18E+03/4.25E+02	9.70E+04/1.15E+04	1.97E+04/6.95E+03	**5.10E-02/1.68E-02**
	Rank	4	2	5	3	**1**
$f2$	Mean/Std	1.34E+05/2.47E+04	3.63E+03/1.47E+03	3.08E+05/2.79E+04	1.24E+05/1.10E+04	**9.99E+02/9.58E+02**
	Rank	4	2	5	3	**1**
$f3$	Mean/Std	8.06E+03/1.30E+03	5.02E+02/3.93E+02	1.98E+04/3.36E+03	2.40E+03/5.37E+02	**4.26E-01/1.38E-01**
	Rank	4	2	5	3	**1**
$f4$	Mean/Std	6.05E+03/1.50E+03	5.13E+02/8.41E+01	1.37E+04/3.43E+03	2.26E+03/5.99E+02	**2.53E+02/5.81E+01**
	Rank	4	2	5	3	**1**
$f5$	Mean/Std	2.13E+01/2.49E-02	2.13E+01/2.96E-02	2.13E+01/3.04E-02	**2.11E+01/3.64E-02**	2.13E+01/3.60E-02
	Rank	4	2	3	**1**	5
$f6$	Mean/Std	1.09E+02/8.19E+00	9.23E+01/8.74E+00	1.20E+02/5.52E+00	1.32E+02/2.84E+00	**5.71E+01/1.13E+01**
	Rank	3	2	4	5	**1**
$f7$	Mean/Std	1.64E+03/1.28E+02	8.22E+02/1.05E+02	1.91E+03/1.05E+02	1.96E+03/6.65E+01	**3.56E+02/3.42E+01**
	Rank	3	2	4	5	**1**
$f8$	Mean/Std	1.62E+03/1.84E+02	1.04E+03/1.24E+02	1.81E+03/1.10E+02	2.05E+03/3.20E+01	**4.22E+02/8.46E+01**
	Rank	3	2	4	5	**1**
$f9$	Mean/Std	1.69E+03/1.60E+02	1.16E+03/1.44E+02	1.82E+03/1.84E+02	2.39E+03/7.20E+01	**5.63E+02/9.94E+01**
	Rank	3	2	4	5	**1**
$f10$	Mean/Std	2.08E+04/6.63E+02	1.72E+04/1.66E+03	2.43E+04/5.29E+02	**1.47E+04/2.72E+03**	1.69E+04/9.31E+02
	Rank	4	3	5	**1**	2
$f11$	Mean/Std	4.21E+03/7.29E+02	1.44E+03/1.77E+02	2.53E+03/8.46E+01	2.25E+03/3.23E+01	**7.35E+02/6.65E+01**
	Rank	5	2	4	3	**1**
$f12$	Mean/Std	3.69E+03/5.46E+02	1.71E+03/2.24E+02	2.51E+03/8.96E+01	2.24E+03/2.85E+01	**7.85E+02/5.49E+01**
	Rank	5	2	4	3	**1**
$f13$	Mean/Std	2.15E+04/7.56E+02	1.83E+04/1.04E+03	2.49E+04/5.49E+02	2.06E+04/1.16E+03	**1.71E+04/1.47E+03**
	Rank	4	2	5	3	**1**
$f14$	Mean/Std	4.46E+02/1.87E+01	4.86E+02/1.77E+01	4.69E+02/9.28E+00	3.00E+03/5.00E+02	**3.79E+02/2.19E+01**
	Rank	2	4	3	5	**1**
$f15$	Mean/Std	7.64E+03/4.64E+02	6.72E+03/2.04E+03	9.09E+03/6.57E+02	2.12E+04/3.61E+02	**2.21E+03/2.06E+02**
	Rank	3	2	4	5	**1**
Count		0	0	0	2	**13**
Ave Rank		3.67	2.2	4.27	3.53	**1.33**
Total Rank		4	2	5	3	**1**

维度:80，以上数据独立运行 10 次的均值

4.1　目标函数

目标函数为

$$F(X)=\lambda_1 f_1(x)+\lambda_2 f_2(x) \tag{21}$$

其中：λ_1 为动力性发挥程度加权因子；λ_2 为经济性加权因子；$f_1(x)$ 为动力性分目标函数；$f_2(x)$ 为经济性分目标函数。选择 $0<\lambda_1<\lambda_2<1$，取加权因子 λ_1=0.2，$\lambda_2=0.8$。

动力性分目标函数：

$$f_1(x)=\int_0^u dt=\int_0^u \frac{\delta\times m}{(F_t-F_f-F_w)}du \tag{22}$$

$\delta=1.06+0.04 i_g^{\ 2}$，$F_t, F_f, F_w$ 分别为汽车的驱动力、滚动阻力和空气阻力。

经济性分目标函数(在速度 u_a 下行驶某段距离 ΔS 的耗油量)：

$$\Delta Q=\frac{K\times P_e\times g_e\times(n_e,P_e)}{102\times u_a\times\rho}\times\Delta S \tag{23}$$

其中，ρ 为燃油重度(N/L)，取值为 7.0。K 为加权系数，等速时取 1，加速时取 1.05；$u_a\in[10,20,30,40,50]$ 为汽车行驶加速(km/h)。发动机的燃油消耗率变化如图 7 所示。

表 6　Friedman 检验结果

Tab. 6　Friedman test results

维度(D)	P-value	FPA	TMFPA	HLFPA	ISSA	MSFPA
30D	5.3308E-11	3.2	2	4.2	4.53	1
80D	5.8656E-7	3.67	2.2	4.27	3.53	1.33

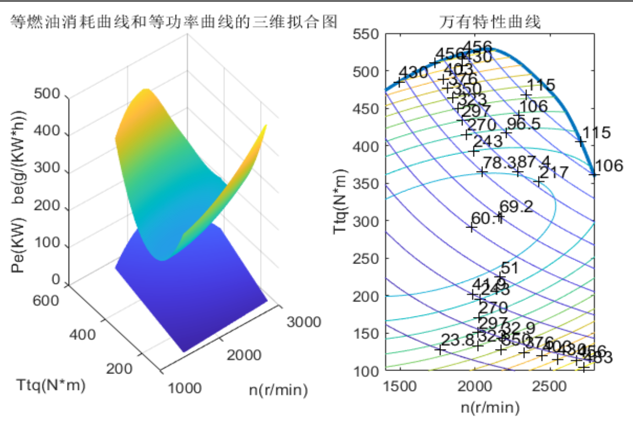


图 7　燃油消耗率变化

Fig. 7　Change of fuel consumption rate

4.2　参数优化约束条件

在燃油汽车优化满足动力性需要，之后满足燃油经济性[25]：

$$g_1(X)=u_{al}-u_{a\max}\le 0 \tag{24}$$

其中，$u_{al}\in[0,50]$，汽车行驶车速(km/h)。

$$g_2(X)=i_l-i_{\max}\le 0 \tag{25}$$

其中，i_l 为汽车最大爬坡度要求下的限值。

$$g_3(X)=D_l-D_{l\max}\le 0 \tag{26}$$

其中，D_l 为汽车最大动力因数要求的下限值。

$$g_4(X)=D_{nl}-D_{n\max}\le 0 \tag{27}$$

其中，D_{nl} 为汽车最高挡的最大动力因数要求的下限值。

$$g_5(X)=\frac{T_{tq\max} i_{g1} i_{g0}\eta_T}{r}-F_{z\varphi}\varphi\le 0 \tag{28}$$

其中，$T_{tq\max}$=132，发动机最大转矩(N*m); $F_{Z\varphi}=G/4$，驱动轮上的法向反作用力(N); φ=0.7 为地面附着系数。

$$g_6(X)=0.85\times q-i_{g1}/i_{g2}\le 0 \tag{29}$$

其中，公比 $q=\sqrt[5]{i_{g1}/i_{g5}}$。其他约束如下所示。

$$g_7(X)=i_{g1}/i_{g2}-1.20q\le 0 \tag{30}$$

$$g_8(X)=0.80q-i_{g2}/i_{g3}\le 0 \tag{31}$$

$$g_9(X)=i_{g2}/i_{g3}-1.10q\le 0 \tag{32}$$

$$g_{10}(X)=0.75q-i_{g3}/i_{g4}\le 0 \tag{33}$$

$$g_{11}(X)=i_{g3}/i_{g4}-1.05q\le 0 \tag{34}$$

$$g_{12}(X)=0.70q-i_{g4}/i_{g5}\le 0 \tag{35}$$

$$g_{13}(X)=i_{g4}/i_{g5}-1.0q\le 0 \tag{36}$$

$$g_{14}(X)=i_{g2}/i_{g3}-0.95 i_{g1}/i_{g2}\le 0 \tag{37}$$

$$g_{15}(X)=i_{g3}/i_{g4}-0.95 i_{g2}/i_{g3}\le 0 \tag{38}$$

$$g_{16}(X)=i_{g4}/i_{g5}-0.95 i_{g3}/i_{g4}\le 0 \tag{39}$$

$$g_{17}(X)=X_l-X\le 0 \tag{40}$$

$$g_{18}(X)=X-X_h\le 0 \tag{41}$$

其中，X_h和X_l 为变速器和主减速器传动比上下限值构成的向量。

4.3　汽车传动参数优化实验结果分析

在算法性能的基础上，对多策略的花授粉算法(MSFPA)在汽车动力传动参数优化问题，结合之前的测试算法进行综合对比。以此表明该算法在实际应用问题上的可适用性。具体的目标函数和约束条件已在第二节中有详细描述。对比结果如图 8 所示，具体优化后的值见表 7。

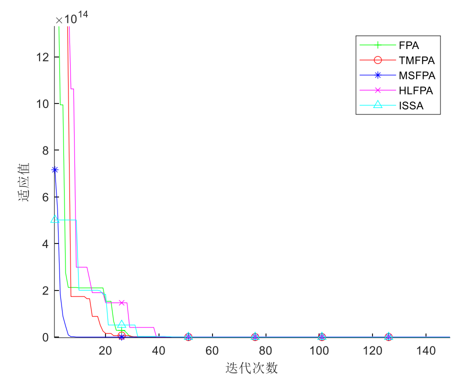


图 8　汽车传动参数各函数适应值对比

Fig. 8　Comparison of adaptive values of functions of automobile transmission parameters

图 8 为了方便展示放大了前 70 次迭代部分，从图中可以看出，MSFPA 算法的收敛速度最快，仅在第 5 次迭代左右就已经到达最低点，而 fpa 算法虽然初始时是最低的，但是后续的收敛速度明显很慢。而其他的算法不管是收敛精度还是在收敛速度上都逊色于 MSFPA 算法。

根据表 7 优化后参数的结果可以看出，5 种对比算法最后适应值最小的是 MSFPA 为 374，其他算法略大于该算法。而汽车传动系的变速箱的速比和主减速器的速比 $X_1\sim X_6$，其值都优于对比算法。因此 MSFPA 算法在汽车动力传动系统参数优化方面相较于对比算法有一定的优越性，在这一类问题上具有参考价值。

表 7　优化参数详情

Tab. 7　Optimization parameters details

算法	X_1	X_2	X_3	X_4	X_5	X_6	$f_{\min}$
FPA	1.28E+00	1.00E+00	8.34E-01	7.44E-01	7.31E-01	2.16E+00	3.76E+02
MSFPA	4.54E-01	3.70E-01	3.22E-01	3.02E-01	3.02E-01	7.27E-01	3.74E+02
TMFPA	1.56E+00	1.00E+00	6.78E-01	5.00E-01	4.43E-01	2.11E+00	3.75E+02
HLFPA	1.28E+00	1.03E+00	9.23E-01	8.77E-01	8.77E-01	2.35E+00	3.77E+02
ISSA	1.53E+00	1.00E+00	6.89E-01	5.00E-01	4.56E-01	3.00E+00	3.75E+02

5 结束语

为了改善 MSFPA 算法的收敛速度，收敛精度以及容易陷入局部最优解的问题，提出一种结合笛卡尔积透镜成像和随机邻域变异策略的花授粉算法，利用透镜成像策略增加解的多样性帮助算法跳出局部最优解，引入随机邻域变异策略加强算法的局部搜索能力，最后根据实验结果对比 4 种不同的算法，MSFPA 算法对各类单模态、多模态、复合、低维、高维都具有更好的搜索精度和收敛速度，最后结合汽车传动参数优化问题进一步验证 MSFPA 算法的实际应用能力，结果表明该算法具有一定的研究意义，为求解复杂问题提供了一种新的思路和解决办法。

参考文献：

[1] Yang X S. Flower pollination algorithm for global optimization [J]. Lecture Notes in Computer Science, 2012, 7445 (1): 240-249.

[2] Singh P, Mittal N. An efficient localization approach to locate sensor nodes in 3D wireless sensor networks using adaptive flower pollination algorithm [J]. Wireless Networks, 2021, 27 (3): 1999-2014.

[3] Mergos P E. Optimum design of 3D reinforced concrete building frames with the flower pollination algorithm [J]. Journal of Building Engineering, 2021, 44: 102935.

[4] Ja Yaraman K K, Srilakshmi K, Ja Yaraman S. Modified Flower pollination-based segmentation of medical images [J]. International Journal of Image and Graphics, 2021: 1.

[5] Mohanty S, Dash R. A flower pollination algorithm based Chebyshev polynomial neural network for net asset value prediction [J]. Evolutionary Intelligence, 2021.

[6] Sakib N, Kabir M W U, Rahman M S, *et al.* A comparative study of flower pollination algorithm and bat algorithm on continuous optimization problems [J]. International Journal of Applied Information Systems, 2014, 7 (9) .

[7] Nabil E. A modified flower pollination algorithm for global optimization [J]. Expert Systems with Applications, 2016, 57 (C): 192-203.

[8] Xiao Huihui, Wan Changxuan, Duan Yanming, *et al.* Flower pollination algorithm based on simulated annealing [J]. Journal of Computer Applications, 2015, 35 (4): P1062-1066, 1070.

[9] Zhou Yongquan, Wang Rui, Luo Qifang. Elite opposition-based flower pollination algorithm [J]. Neurocomputing, 2016, 188: 294-310.

[10] Ramadas M, Pant M, Abraham A, *et al.* ssFPA/DE: an efficient hybrid differential evolution–flower pollination algorithm based approach [J]. International Journal of System Assurance Engineering and Management, 2018, 9 (1): 216-229.

[11] 肖辉辉，万常选，段艳明，等. 基于引力搜索机制的花朵授粉算法 [J]. 自动化学报, 2017, 43 (4): 19. (Xiao Huihui, Wan Changxuan, Duan Yanming, *et al.* Flower pollination algorithm based on gravity search mechanism [J]. Acta Automatica Sinica, 2017, 43 (4): 576-594)

[12] 卞京红，贺兴时，杨新社. 基于萤火虫算法的自适应花授粉优化算法 [J]. 计算机工程与应用, 2016, 52 (21): 162-167, 217. (Bian Jinghong, He Xingshi, Yang Xinshe. Hybrid algorithm of firefly algorithm and self-adaptive flower pollina-tion algorithm [J]. Computer Engineeringand Applications, 2016, 52 (21): 162-167, 217)

[13] Dubkov A, Spagnolo B, Uchaikin V V. Levy flight superdiffusion: an introduction [J]. International Journal of Bifurcation and Chaos, 2008, 18 (9): 2649-2672.

[14] 郑永爱，宣蕾. 混沌映射的随机性分析 [J]. 计算机应用与软件, 2011, 28 (12): 274-276, 292. (Zheng Yongai, Xuan Lei. Analysing randomicity of chaos mapping [J]. Computer Applications and Software, 2011, 28 (12): 274-276, 292)

[15] 高鹰，谢胜利. 基于模拟退火的粒子群优化算法 [J]. 计算机工程与应用, 2004, 40 (1): 47-50. (Gao Ying, Xie Shengli. Particle sswarm optimization algorithms based on simulated annealing [J]. Computer Engineering and Applications, 2004, 40 (1): 47-50)

[16] Price K, Storn R. Differential evolution [J]. Dr. Dobb's Journal, 1997, 22 (4): 18-23.

[17] Storn R, Price K. Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces [J]. Journal of Global Optimization, 1995, 23 (1) .

[18] Das S, Abraham A, Chakraborty U K, *et al.* Differential evolution using a neighborhood-based mutation operator [J]. IEEE Trans on Evolutionary Computation, 2009, 13 (3): 526-553.

[19] Peng Hu, Guo Zhaolu, Deng Changshou, *et al.* Enhancing differential evolution with random neighbors based strategy [J]. Journal of Computational Science, 2018, 26: 501-511.

[20] 呼忠权，王洪斌. 基于 Lévy 飞行的自适应差分进化算法 [J]. 现代电子技术, 2020, 43 (4): 167-172. (Hu Zhongquan, Wang Hongbin. Adaptive differential evolution algorithm based on Lévy flight [J]. Modern Electronics Technique, 2020, 43 (4): 167-172)

[21] 洪露，贺兴时，杨新社. 基于三重动态调整的花授粉算法 [J]. 西安工程大学学报, 2021, 35 (2): 97-103. (Hong Lu, He Xingshi, Yang Xinshe. The flower pollination algorithm based on triple dynamic adjustment [J]. Journal of Xi’an Polytechnic University, 2021, 35 (2): 97-103)

[22] 宁杰琼，何庆. t-分布扰动策略和变异策略的花授粉算法 [J]. 小型微型计算机系统, 2021, 42 (1): 64-70. (Ning Jieqiong, He Qing. Flower pollination algorithm based on t-distribution perturbation strategy and mutation strategy [J]. Journal of Chinese Computer Systems, 2021, 42 (1): 64-70)

[23] 毛清华，张强. 融合柯西变异和反向学习的改进麻雀算法 [J]. 计算机科学与探索, 2021, 15 (6): 1155-1164. (Mao Qinghua, Zhang Qiang. Improved sparrow algorithm combining cauchy mutation and opposition-based learning [J]. Journal of Frontiers of Computer Science & Technology, 2021, 15 (6): 1155-1164)

[24] Elmazi D, Oda T, Sakamoto S, *et al.* Friedman test for analysing WMNs: a comparison study for genetic algorithms and simulated annealing [C]// Proc of the 9th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing. 2015.

[25] 李军民，王俊昌. 基于遗传算法矿用汽车动力传动系参数优化设计 [J]. 机械设计与制造, 2018, (11): 228-232. (Li Junmin, Wang Junchang. Parameters optimization design of powertrain system of mining truck based on genetic algorithm [J]. Machinery Design & Manufacture, 2018, (11): 228-232)